11. Pankaj

Program Name: Pankaj.java **Input File: pankaj.dat**

Pankaj will be working for the Ultimate Intelligent Learners Summer camp in the summer of 2022. This camp is geared towards getting the campers to make new friends within their own community and the surrounding areas. This means there is the potential for a camper to know zero, one, or more than one of the other campers attending any given session.

The camp counselors have asked Pankaj to take a list of campers as well as a list of known relationships, and determine if the campers can be broken into two distinct groups such that no two campers in a group have a relationship with anyone else in the group. Remember, the focus of this camp is to get the campers to make new friends, and how can you make new friends with people you already know?

Can you help Pankaj write a program to determine whether it is possible or impossible to break a group of campers up into two distinct groups such that no two campers know anyone else in their group?

Input: The input will consist of an integer *I*, the number of test cases. *I* will be in the range of *[1,20]*. For each test case, input will consist of two lines. Line 1 will contain the name of all the campers attending a given session. Names will be limited to first names only, and only one word first names. It will be guaranteed that no two campers have the same exact first name. Names in the list will be separated by a comma "," and there will be no spaces in the list. Line 2 will consist of the known relationships between the campers. The relationships will be given in the form: "`camper1<->camper2`" this means that `camper1` knows `camper2` and `camper2` knows `camper1`. A relationship in this problem is always two-way, each camper will know the other camper if a relationship is present. Relationships will be separated by a comma "," and there will be no spaces in the list. There will be at least one relationship present between two campers, but potentially more.

Output: For each test case, you are to output "Test Case #: possible" if the campers can be broken into two distinct groups such that know two campers know each other or "Test Case#: impossible" if the campers can not be broken into two such groups.

Sample input:

```
5
Alex,Brent,Chris,Dave,Eric,Fred
Alex<->Chris,Alex<->Eric,Brent<->Dave,Brent<->Fred,Chris<->Eric,Dave<->Fred
Alex,Brent,Chris,Dave,Eric,Fred
Alex<->Chris,Alex<->Eric
Alex,Brent,Chris,Dave,Eric,Fred
Alex<->Brent,Brent<->Chris,Chris<->Dave,Dave<->Eric,Eric<->Fred,Fred<->Alex
Alex,Brent,Chris,Dave,Eric,Fred
Alex<->Brent,Alex<->Chris,Alex<->Dave,Alex<->Eric,Alex<->Fred
Alex,Brent,Chris,Dave,Eric,Fred
Alex<->Brent,Alex<->Chris,Alex<->Dave,Alex<->Eric,Alex<->Fred,Brent<->Chris
```

Sample output:

```
Test case 1: impossible
Test case 2: possible
Test case 3: possible
Test case 4: possible
Test case 5: impossible
```